

Задача А. Message Relay

Имя входного файла: `relay.in`
Имя выходного файла: `relay.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

n ($1 \leq n \leq 1000$) коров Фермера Джона последовательно пронумерованы от 1 до n . Коровы хотят переговариваться друг с другом с помощью жестяных банок и проводов.

Каждая корова может послать сообщение не более чем одной другой корове: для коровы i значение $F(i)$ указывает Вам индекс коровы, которой корова i может послать любое сообщение, которое она получит (это число всегда отличается от i). Если $F(i)$ равно 0, то эта корова не может переслать сообщение никому.

К несчастью, существует возможность что сообщение посланное некоторой коровой, будет ходить по кругу бесконечно. Корова называется «циклической», если сообщение, посланное этой коровой, попадает в бесконечный цикл. Коровы хотят избежать посылок сообщений от циклических коров. Пожалуйста, помогите им, подсчитав общее количество нециклических коров.

Формат входных данных

Первая строка содержит количество коров, n ($1 \leq n \leq 1000$).

Следующие n строк содержат $F(i)$. Строка $i + 1$ содержит значение $F(i)$

Формат выходных данных

Выведите общее количество нециклических коров.

Примеры

relay.in	relay.out
5	2130567170
0	
4	
1	
5	
4	

Замечание

Корова 1 нециклическая, поскольку она не пересылает сообщения.

Корова 3 также нециклическая, поскольку она пересылает сообщения корове 1, которая не пересылает сообщений.

Все остальные коровы циклические.

Задача В. Трактор

Имя входного файла: `tractor.in`
Имя выходного файла: `tractor.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Одно из полей Фермера Джона весьма холмисто. И он хочет купить новый трактор для работы на этом поле. Поле описывается решеткой из $n \times n$ неотрицательных целых высот ячеек. Трактор может перемещаться из ячейки в соседнюю (на один шаг на север, юг, запад или восток), с разницей их высот d ровно за d единиц денег.

ФД хочет заплатить достаточно, так чтобы его трактор, начиная с некоторой ячейки поля мог посетить как минимум половину ячеек поля. Если число ячеек в поле — нечетное, то половина, округленная вверх.

Определите минимальную стоимость покупки трактора способного выполнить эту задачу.

Формат входных данных

Первая строка содержит число n — размер фермы Джона ($1 \leq n \leq 500$). Следующие n строк содержат описание поля. Каждая строка содержит n разделенных одиночными пробелами неотрицательных целых чисел (каждое не более миллиона), определяющих строку поля ФД.

Формат выходных данных

Выведите минимальную стоимость трактора, который способен объехать не менее половины этого поля.

Пример

tractor.in	tractor.out
5 0 0 0 3 3 0 0 0 0 3 0 9 9 3 3 9 9 9 3 3 9 9 9 9 3	3

Задача С. Доим коров

Имя входного файла: `msched.in`
Имя выходного файла: `msched.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

n коров Фермера Джона пронумерованы последовательно от 1 до n . Для доения коровы i требуется t_i единиц времени. Однако некоторых коров необходимо подоить ранее других (из-за их положения на ферме). Если корову A требуется подоить перед коровой B , ФД должен полностью закончить дойку коровы A , прежде чем начать дойку коровы B .

Для того, чтобы подоить всех своих коров как можно быстрее, ФД нанял большое количество доярок — достаточное для того, чтобы доить любое количество коров одновременно.

Определите минимальное количество времени, требуемое для дойки всех коров.

Формат входных данных

Первая строка содержит n m — количество коров и количество ограничений, соответственно ($1 \leq n \leq 10\,000, 1 \leq m \leq 50\,000$). Следующие n строк, содержат t_i — времена, требуемые для дойти коров. Следующие m строк содержат числа a_i и b_i — ограничения на дойку ($1 \leq a_i, b_i \leq n, a_i \neq b_i$).

Формат выходных данных

Выведите минимальное количество времени, требуемое чтобы подоить всех коров.

Пример

<code>msched.in</code>	<code>msched.out</code>
3 1 10 5 6 3 2	11

Задача D. Разделение фермы

Имя входного файла: `partition.in`
Имя выходного файла: `partition.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ферма Джона разделена на $n \times n$ квадратных пастбищ ($2 \leq n \leq 15$). Снаружи есть изгородь, но между пастбищами коровы могут переходить свободно.

ФД решил построить изгороди, чтобы отделить коров друг от друга. Каждая изгородь может быть горизонтальной или вертикальной через всю ферму, и изгороди не могут проходить через пастбища. По финансовым соображениям ФД может построить не более чем k изгородей ($1 \leq k \leq 2 \cdot n - 2$).

ФД хочет построить изгороди так, чтобы минимизировать размер наибольшей из получившихся в результате групп коров (две коровы находятся в одной группе, если они могут посетить друг друга, не пересекая никакую изгородь).

По заданным количествам коров на пастбищах, вычислите размер наибольшей группы коров, если ФД построит изгороди оптимально.

Формат входных данных

Первая строка входных данных содержит два целых числа, n и k .

Следующие n строк содержат n чисел на каждой строке, описывающих количество коров в каждом пастбище. На каждом пастбище не менее 0 и не более 1000 коров.

Формат выходных данных

Выведите минимально возможный размер наибольшей группы коров.

Пример

partition.in	partition.out
3 2 1 1 2 1 1 2 2 2 4	4

Замечание

ФД должен построить изгороди между колонками 2 и 3 и между строками 2 и 3. В результате получится 4 группы по 4 коровы в каждой.

Задача Е. Taxi

Имя входного файла: `taxi.in`
Имя выходного файла: `taxi.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Беси открыла такси-сервис для других коров на ферме. Коровы собрались в различных местах вдоль изгороди длины m ($1 \leq m \leq 1\,000\,000\,000$) и каждая хочет переместиться в некоторое другое место вдоль изгороди. Беси должна подобрать корову в том месте, где она находится и отвезти в то место, куда она хочет.

Автомобиль Беси маленький и за раз может возить только одну корову. Коровы могут входить машину и выходить из нее мгновенно.

Беси хочет минимизировать расстояние проезда. Вам даны стартовые и финишные позиции n коров ($1 \leq n \leq 100\,000$), определите минимальное количество езды, которое должна выполнить Беси. Беси поняла, что иногда выгодно высаживать корову не в позиции ее назначения.

Беси начинает в самой левой точке изгороди - позиции 0 и должна закончить свое путешествие в самой правой точке - в позиции m .

Формат входных данных

Первая строка содержит два целых числа: n и m . Следующие n строк содержат по два целых числа: s_i и t_i ($0 \leq s_i, t_i \leq m$), указывающих стартовую и конечную позиции i -ой коровы.

Формат выходных данных

Одно целое число, указывающее общее расстояние, которое проедет Беси. Заметим, что результат может не поместиться в 32-битное целое.

Пример

<code>taxi.in</code>	<code>taxi.out</code>
2 10 0 9 6 5	12

Задача F. Тур по Амазонке

Имя входного файла: `route.in`
Имя выходного файла: `route.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Беси открыла агентство путешествий вдоль реки Амазонка. Имеется несколько туристических мест на обеих сторонах реки, для каждого места некоторое целое число задает насколько это место интересно туристам.

Туристические места соединены маршрутами через реку. Иными словами, нет маршрутов, соединяющих туристические места по одной стороне реки.

Беси хочет найти такую последовательность посещения туристических мест, чтобы максимизировать сумму значений ассоциированных с каждым посещенным местом и чтобы никакие два маршрута в этой последовательности не пересекались. Два маршрута $(a \leftrightarrow x)$ и $(b \leftrightarrow y)$ пересекаются тогда и только тогда, когда $(a < b \text{ и } y < x)$ или $(b < a \text{ и } x < y)$ или $(a = b \text{ и } x = y)$.

Помогите Беси найти такой маршрут. Беси может начинать и заканчивать маршрут в любом месте и на любой стороне реки.

Формат входных данных

Первая строка входных данных содержит три, разделенных пробелом, целых числа n ($1 \leq n \leq 40\,000$), m ($1 \leq m \leq 40\,000$), и r ($0 \leq r \leq 100\,000$) указывающих количество мест на левой стороне реки, количество мест на правой стороне реки и количество маршрутов, соответственно.

Следующие n строк содержат по одному целому числу l_i — интересность места i на левой стороне реки ($0 \leq l_i \leq 40\,000$).

Следующие m строк содержат по одному целому числу r_i — интересность места i на правой стороне реки ($0 \leq r_i \leq 40\,000$).

Последние r строк содержат по два целых числа u ($1 \leq u \leq n$) и v ($1 \leq v \leq m$), указывающих двунаправленный маршрут между местом u на левой стороне реки и местом v на правой стороне реки.

Формат выходных данных

Выведите одно целое число, указывающее максимальную сумму значений включенных в тур.

Пример

route.in	route.out
3 2 4 1 1 5 2 2 1 1 2 1 3 1 2 2	8